

Project	<b>IEEE 802.20 Working Group on Mobile Broadband Wireless Access</b> < <a href="http://grouper.ieee.org/groups/802/20/">http://grouper.ieee.org/groups/802/20/</a> >	
Title	<b>LDPC Code Proposal – Technology Specification</b>	
Date Submitted	<b>2007-03-05 (March 5, 2007)</b>	
Sources	Sung-Eun Park, Seunghoon Choi Samsung Electronics, Suwon, Korea	Email : { <a href="mailto:se.park@samsung.com">se.park</a> , <a href="mailto:seunghoon.choi@samsung.com">seunghoon.choi</a> } <a href="mailto:se.park@samsung.com">@samsung.com</a>
	Thierry Lestable Samsung Electronics Research Institute, UK	Email : <a href="mailto:thierry.lestable@samsung.com">thierry.lestable@samsung.com</a>
	Anna Tee Samsung Telecommunications America	Voice: 1 (972) 761-7437 Email: <a href="mailto:atee@sta.samsung.com">atee@sta.samsung.com</a>
Re:	IEEE 802.20 Call for Proposals	
Abstract	This document proposes an LDPC coding scheme for Mobile Broadband Wireless Access Systems.	
Purpose	To provide detail specification text for LDPC coding in 802.20 standard draft.	
Notice	This document has been prepared to assist the IEEE 802.20 Working Group. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.	
Release	The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE's name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE's sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by IEEE 802.20.	
Patent Policy	The contributor is familiar with IEEE patent policy, as outlined in <a href="#">Section 6.3 of the IEEE-SA Standards Board Operations Manual</a> < <a href="http://standards.ieee.org/guides/opman/sect6.html#6.3">http://standards.ieee.org/guides/opman/sect6.html#6.3</a> > and in <i>Understanding Patent Issues During IEEE Standards Development</i> < <a href="http://standards.ieee.org/board/pat/guide.html">http://standards.ieee.org/board/pat/guide.html</a> >.	

## **Introduction**

The current version of 802.20 air interface standard draft [1] supports two basic channel coding schemes, namely, a rate 1/5 Parallel Concatenated Convolutional Code (PCCC) i.e. Turbo codes and a rate 1/3 Convolutional Code (CC). The rate 1/5 Turbo codes shall be used for number of information bits  $k$  larger than 128, while the rate 1/3 Convolutional Code shall be used for values of  $k$  less than or equal to 128. We propose hereafter to include an optional Low-Density Parity-Check (LDPC) coding scheme for high data rates. The proposed LDPC codes offer both efficient support of Type II HARQ (Incremental Redundancy) together with similar or better performances than Turbo codes through all HARQ retransmissions. Besides, this proposed code structure enables highly parallelizable decoder architectures, thus resulting in high-throughput decoder implementations.

LDPC codes are fully defined by their sparse parity-check matrices, and can also be represented by a Tanner Graph, a bipartite graph illustrating the constraints of the code. Such graph consists first of two types of nodes namely variable nodes and check nodes, then of edges which connect those two types of nodes. Variable nodes represent bits in the codeword, but they can also contain some punctured bits that are not transmitted on the channel. Check nodes represent the constraints that define the code: the set of variable nodes connected to any given check node is constrained to sum to zero.

LDPC codes can be decoded by Message-Passing based algorithms. One of them is the Pearl's Belief-Propagation (BP) algorithm which passes beliefs in the form of Log-Likelihood Ratios (LLR) along the edges of the bipartite graph. Although optimal only for tree structure codes (cycle free), near optimal performances are usually obtained. Moreover, the complexity of BP algorithm is proportional to the number of edges in the graph. Due to the sparseness of the parity-check matrix, and thus of the corresponding bipartite graph, the resulting decoding complexity is quite affordable.

## **Features of the proposed LDPC code**

The proposed LDPC code has excellent performance, and contains features that provide packet length flexibility whilst maintaining low encoding and decoding complexity. Besides, it enables the support of Incremental Redundancy (IR) HARQ.

- **Structured LDPC code.** The entire parity-check matrix (i.e., both the sections that correspond to the information and parity bits) is composed of the same style of blocks which are either cyclic permutation matrices or zero matrices. This reduces decoder implementation complexity and allows structured decoding.
- **Code length flexibility.** The shift values for each block size are derived from the matrix of the largest allowable block size. This facilitates compact representation of parity-check matrices, thus resulting in memory storage savings, together with efficient implementations of the encoder and decoder parts.
- **Multi-Edge-Type LDPC code.** Parity-check matrix contains several types of edges including punctured information nodes and degree-one parity nodes. The punctured information nodes give higher noise thresholds with lower degrees, resulting in performance and complexity improvement. The degree-one parity nodes are very convenient for HARQ
- **Efficient Encoding.** The combination of Richardson & Urbanke's encoding algorithm and simple single parity-check coding facilitates efficient encoding.
- **HARQ support.** First, the lowest code-rate multi-edge-type parity-check matrix is designed to support HARQ. Then only a part of codeword (puncturing) is transmitted during each HARQ transmission (Incremental Redundancy). In other words, higher rate codes are achieved by puncturing parity bits from the lower rate codes. As a consequence, such LDPC codes are thus Rate-Compatible Punctured Codes (RCPC), suitable for IR Type II HARQ.

### **Recommended Text Changes:**

Modify the text in IEEE 802.20/D2.1 as follows, adjusting the section numbering as required:

<Replace the contents of section 9.2.2 (p.559 line 24 to 26) with the following text.>

The air interface shall support 2 mandatory coding schemes, namely a rate 1/5 parallel turbo code and a rate 1/3 convolutional code, together with an optional Rate Compatible LDPC Code. The LDPC code shall be used for high data rates (values of  $k$  larger than 512), and the rate 1/5 turbo code shall be used for values of  $k$  larger than 128, while the rate 1/3 convolutional code shall be used for values of  $k$  less than or equal to 128.

<Add the following section to the end of section 9.2.2.2.>

#### **9.2.2.3 LDPC encoding**

The core LDPC codes are systematic structured LDPC codes. The parity-check matrix of the structured LDPC code is lifted from a small base parity-check matrix of size  $m \times n$ . Each non-zero element in the base parity-check matrix is replaced by an  $L \times L$  cyclic permutation matrix which can be expressed by the powers of a matrix  $P$  defined by

$$P = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

A base model matrix is defined for the lowest code rate that enables support of Incremental Redundancy (IR) HARQ. One parity-check matrix is used for each packet format defined at Table 73 in section 7.5.6.7 and Table 95 in section 7.7.6.8 respectively. Variable code lengths are supported by increasing or decreasing the size of cyclic permutation matrix. Assuming that  $(i, j)$ -th element in base matrix is non-zero. Then shift value  $p(f, i, j)$  corresponding to the expansion factor  $L_f$  is derived from the original shift value  $p(i, j)$  by using a modulo function  $p(f, i, j) = \text{mod}(p(i, j), L_f)$ .

The parity-check matrix of structured LDPC codes is compactly represented by degrees, positions and shift values of non-zero elements in base matrix. The format for representing parity-check matrix is given in Table 98. Since the LDPC codes are systematic codes, information bits are always located in the left of the parity-check matrix, whereas parity bits are located in the right side. Note that some columns will be punctured before transmission and that position of the punctured columns is always leftmost. To support IR HARQ, only a part of codeword is transmitted during each HARQ transmission serially from left to right of the codeword corresponding to the lowest rate parity-check matrix.

**Table 98 Parity-check matrix format**

<b>Number of columns, Number of rows, Number of punctured columns</b>	
<b>Column degrees</b>	
<b>Row degrees</b>	
<b>Positions of non-zero block in rows</b>	<b>Shift values of non-zero block in rows</b>



Table 99b Parity-check matrix for FL PFI 01, RL PFI 01

48 37 4	
9999999999999322222222222222222111111111111111111111111111111111111111	
55455456444454554444444443333344444444	
0 4 6 11 12	25 15 9 1 0
1 5 7 12 13	16 3 20 0 0
2 10 13 14	34 6 0 0
3 6 9 14 15	23 0 7 0 0
0 5 8 15 16	15 8 3 0 0
1 7 16 17	7 3 2 0 0
2 4 9 17 18	9 2 3 0 0 0
3 5 10 11 18 19	24 0 0 0 0 0
0 8 19 20	1 0 0 0
1 4 20 21	6 29 0 0
2 10 21 22	7 2 0 0
3 9 22 23	5 26 0 0
0 6 7 23 24	20 8 6 0 0
1 8 24 25	20 20 0 0
2 6 7 25 26	0 30 0 0 0
3 4 5 11 26	11 0 0 1 0
8 9 10 27	2 22 25 0
0 5 8 28	29 1 0 0
1 4 7 29	37 7 20 0
2 6 10 30	21 0 14 0
3 5 9 31	0 14 0 0
0 4 6 32	25 7 0 0
1 7 10 33	21 10 19 0
2 8 9 34	0 38 20 0
3 4 35	39 39 0
0 5 36	29 0 0
1 6 37	11 0 0
2 7 38	4 36 0
3 8 39	28 0 0
0 5 9 40	13 0 0 0
1 9 10 41	14 3 0 0
2 5 6 42	24 0 0 0
3 7 8 43	37 21 0 0
0 9 10 44	6 0 6 0
1 4 6 45	0 9 0 0
2 8 10 46	0 24 0 0
3 4 7 47	35 2 20 0

**Table 99c Parity-check matrix for FL PFI 02, RL PFI 02**

70 59 4	
13 13 14 14 11 12 12 11 12 12 12 3 2 1	
1 1 1 1 1 1 1 1 1 1 1 1 1	
4 3 3 3 3 4 3 4 3 3 3 3 3 4 4 4 3 3 3 3 4 3 4 3 3 4 4 4 4 5 4	
0 8 11 12	43 3 1 0
4 12 13	79 0 0
1 13 14	52 0 0
9 14 15	29 0 0
5 15 16	18 0 0
2 10 16 17	18 5 0 0
6 17 18	53 0 0
3 7 18 19	78 11 0 0
8 19 20	60 0 0
0 20 21	72 0 0
5 21 22	32 0 0
10 22 23	10 0 0
1 6 23 24	20 64 0 0
3 11 24 25	1 0 0 0
0 4 25 26	30 41 0 0
7 26 27	7 0 0
2 27 28	48 0 0
8 28 29	0 0 0
9 29 30	4 0 0
1 4 30 31	35 0 0 0
6 31 32	0 0 0
2 5 32 33	3 20 0 0
7 33 34	0 0 0
9 34 35	0 0 0
3 10 11 35	60 0 1 0
0 1 2 36	78 5 7 0
1 2 3 37	48 75 1 0
0 2 3 38	55 7 60 0
0 1 3 39	3 16 0 0
0 6 7 9 40	66 19 0 12 0
1 8 10 41	54 77 0 0
2 4 9 42	59 26 0 0
3 4 5 43	0 37 73 0
0 7 10 44	63 74 7 0
1 4 10 45	8 16 0 0
2 5 10 46	25 77 0 0
3 6 10 47	74 3 0 0
0 8 9 48	0 19 0 0
1 5 8 49	64 9 0 0
2 4 7 50	2 31 0 0
3 4 8 51	68 48 0 0
0 5 6 52	67 0 5 0
1 7 9 53	50 0 17 0
2 6 8 54	14 1 0 0
3 5 7 55	0 68 0 0
2 5 9 56	69 0 14 0
3 6 9 57	0 35 0 0
0 4 10 58	0 27 0 0
1 5 6 59	0 74 18 0
2 6 7 60	0 30 0 0
3 5 7 61	4 51 49 0
0 5 9 62	16 39 0 0
1 7 10 63	0 43 0 0
2 4 8 64	0 15 0 0
3 8 10 65	23 0 15 0
0 6 8 66	0 59 0 0
1 8 9 67	0 0 40 0
2 6 10 68	0 52 0 0
3 4 9 69	0 0 40 0

**Table 99d Parity-check matrix for FL PFI 03, RL PFI 03**

95 80 5			
16 17 17 16 16 13 13 13 13 13 13 13 13 3 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1			
4 4 4 3 4 4 3 4 4 3 4 5 4 4 3 4 4 4 4 3 4 4 4 4 4 5 5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 4 4 4 4 4 4 4			
4 4 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4			
0 8 15 16	3 6 10 55	57 5 1 0	0 21 0 0
6 7 16 17	4 6 12 56	47 48 0 0	15 58 0 0
1 10 17 18	0 7 9 57	60 5 0 0	0 64 0 0
8 18 19	1 6 14 58	15 0 0	16 66 0 0
3 9 19 20	2 7 14 59	17 46 0 0	0 33 70 0
2 13 20 21	3 13 14 60	4 7 0 0	35 34 0 0
12 21 22	4 8 9 61	14 0 0	37 78 0 0
4 11 22 23	0 8 11 62	14 1 0 0	0 18 0 0
3 8 23 24	1 8 12 63	44 0 0 0	13 42 0 0
14 24 25	2 8 13 64	15 0 0	38 0 22 0
0 5 25 26	3 9 14 65	51 57 0 0	39 69 0 0
4 5 15 26 27	4 10 13 66	0 24 0 0 0	12 0 40 0
9 12 27 28	4 11 12 67	6 5 0 0	4 0 0 0
0 14 28 29	1 4 6 13 68	8 7 0 0	2 45 61 0 0
6 29 30	2 5 7 12 69	60 0 0	16 73 0 0 0
2 13 30 31	3 8 11 14 70	19 29 0 0	0 1 23 0 0
1 12 31 32	0 7 10 71	7 72 0 0	0 74 2 0
7 10 32 33	1 8 11 72	54 5 0 0	0 61 5 0
4 11 33 34	2 9 12 73	32 11 0 0	18 81 0 0
9 34 35	3 10 13 74	0 0 0	6 0 0 0
1 6 35 36	4 11 14 75	18 0 0 0	0 56 0 0
3 7 36 37	0 5 12 76	0 0 0 0	35 80 0 0
5 13 37 38	1 6 13 77	0 0 0 0	43 1 0 0
10 14 38 39	2 6 7 78	59 0 0 0	3 25 0 0
2 11 15 39	3 5 9 79	0 0 1 0	0 59 0 0
0 1 5 6 40	1 14 80	22 41 32 0 0	61 0 0
1 2 7 8 41	2 13 81	6 44 45 0 0	0 0 0
2 3 9 42	0 10 82	26 34 0 0	6 0 0
3 4 10 43	4 9 83	63 0 0 0	0 0 0
0 4 11 44	8 12 84	53 12 0 0	0 0 0
0 2 12 45	0 8 10 85	39 0 0 0	36 48 10 0
1 3 13 46	1 9 11 86	54 69 0 0	0 78 8 0
2 4 14 47	2 10 12 87	20 44 0 0	39 30 0 0
0 3 5 48	3 11 13 88	0 61 0 0	12 73 0 0
1 4 6 49	4 12 14 89	5 0 0 0	6 0 16 0
0 1 9 10 50	0 5 13 90	51 0 27 0 0	15 79 0 0
2 3 7 11 51	1 6 14 91	27 19 82 0 0	86 54 24 0
0 5 10 52	2 5 7 92	0 14 0 0	2 49 0 0
1 5 11 53	3 6 8 93	0 5 66 0	0 72 0 0
2 5 7 54	4 7 9 94	10 41 0 0	0 47 36 0







Table 99h Parity-check matrix for FL PFI 07, RL PFI 09

102 82 2			
27 27 8 9 8 9 9 8 8 8 8 8 12 13 13 12 12 13 12 13 3 2 1 1			
1 1			
4 4 4 3 4 4 5 5 4 4 4 4 5 5 4 4 4 4 5 5 5 5 5 5 5 4 3 4 3 4 3 5 5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 4 4 4 5 5 5			
5 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 4 4			
0 18 20 21	1 3 16 61	11 19 1 0	0 40 13 0
1 17 21 22	0 4 15 62	94 24 0 0	0 129 112 0
15 19 22 23	1 5 12 63	38 118 0 0	0 26 0 0
16 23 24	0 6 13 64	93 0 0	0 100 80 0
0 12 24 25	1 7 14 65	38 40 0 0	0 168 111 0
11 14 25 26	0 8 15 66	152 12 0 0	0 0 138 0
1 10 13 26 27	1 9 16 67	130 140 76 0 0	0 16 28 0
0 6 8 27 28	0 10 17 68	85 89 35 0 0	0 112 34 0
1 9 28 29	1 11 18 69	60 27 0 0	0 171 112 0
7 18 29 30	3 6 13 16 70	30 0 0 0	105 0 41 0 0
0 5 30 31	4 7 12 17 71	40 42 0 0	60 152 163 0 0
2 19 31 32	0 5 15 18 72	17 0 0 0	0 146 123 21 0
1 3 4 32 33	1 6 14 19 73	4 49 12 0 0	0 154 24 0 0
0 7 13 33 34	0 7 17 74	139 2 36 0 0	0 131 131 0
1 14 34 35	1 8 16 75	0 118 0 0	0 7 22 0
6 16 35 36	9 12 19 76	35 0 0 0	12 24 0 0
0 10 36 37	2 10 13 18 77	54 36 0 0	106 120 17 8 0
4 17 37 38	5 11 14 19 78	104 11 0 0	155 83 144 15 0
1 8 12 38 39	2 15 18 79	42 32 43 0 0	122 0 20 0
3 9 20 39 40	0 3 10 19 80	128 29 0 0 0	0 78 75 156 0
1 13 15 40 41	1 4 11 18 81	99 0 0 0 0	0 143 8 134 0
0 5 14 41 42	8 12 14 17 82	134 81 0 0 0	16 147 22 0 0
1 6 16 42 43	9 13 15 16 83	38 0 3 0 0	56 113 42 0 0
0 4 17 43 44	0 3 18 84	88 38 0 0 0	0 154 59 0
1 11 19 44 45	0 9 16 85	73 3 96 0 0	0 15 20 0
0 2 12 45 46	1 10 17 86	107 106 121 0 0	0 12 136 0
1 15 46 47	0 11 18 87	2 32 0 0	0 16 49 0
8 47 48	1 12 19 88	88 0 0	0 66 28 0
0 10 48 49	0 6 13 89	34 0 0 0	0 125 12 0
5 49 50	1 7 14 90	0 0 0	0 162 65 0
1 18 50 51	0 12 15 91	8 0 0 0	0 22 47 0
9 51 52	1 8 16 92	0 0 0	0 137 98 0
0 2 7 52 53	2 9 17 93	171 0 63 0 0	160 39 99 0
1 3 11 20 53	3 10 13 18 94	0 0 132 1 0	0 51 154 0 0
0 12 15 54	4 11 14 19 95	0 19 103 0	36 125 126 0 0
1 13 16 55	3 5 12 15 96	0 172 34 0	41 168 25 0 0
0 14 17 56	2 6 13 16 97	0 21 137 0	121 24 0 0 0
1 12 18 57	5 7 14 17 98	0 84 154 0	39 14 85 0 0
0 13 19 58	4 8 15 19 99	0 121 98 0	78 66 0 148 0
1 14 17 59	5 13 17 100	0 169 0 0	50 61 0 0
0 2 19 60	6 14 19 101	0 70 0 0	85 67 0 0













Table 99o Parity-check matrix for FL PFI 14

162 107 2			
333388888888888899999999995555555555444444444555555555532222222			
222111111111111111111111			
11			
5454545454766666666667666766666666666666666666644555555555555555			
5544555555555555555555554444444444444444444444444444444			
0 10 20 55 56	0 4 10 18 109	0 0 0 1 0	60 124 74 0 0
5 15 56 57	1 5 11 19 110	0 0 0 0	158 57 146 0 0
1 11 21 57 58	0 6 12 20 111	0 0 0 0 0	149 27 0 66 0
6 16 58 59	1 7 13 21 112	0 0 0 0	146 33 19 0 0
2 12 22 59 60	0 8 10 14 113	0 0 0 0 0	147 10 71 0 0
7 17 60 61	1 9 11 15 114	0 0 0 0	73 34 16 62 0
3 13 23 61 62	0 3 9 17 115	0 0 0 0 0	62 75 82 0 0
8 18 62 63	1 2 10 18 116	0 0 0 0	18 33 34 0 0
4 14 24 63 64	0 5 11 19 117	0 0 0 0 0	103 73 0 25 0
9 19 64 65	1 4 12 20 118	0 0 0 0	149 43 31 0 0
0 29 35 41 47 65 66	0 7 13 21 119	9 3 0 0 0 0 0 0	8 24 0 175 0
26 32 38 44 66 67	1 6 14 120	0 0 0 0 0 0	19 3 0 0
1 30 36 42 67 68	8 15 16 121	3 8 0 0 0 0 0	0 171 2 0
27 33 39 45 68 69	3 13 42 49 122	0 0 0 0 0 0	118 9 88 69 0
25 31 37 43 69 70	4 14 43 50 123	0 0 0 0 0 0	66 139 1 146 0
28 34 40 46 70 71	5 15 44 51 124	0 0 0 0 0 0	80 138 0 172 0
7 9 29 49 71 72	6 16 45 52 125	6 4 9 5 19 0 0 0	144 66 103 0 0
0 26 48 51 72 73	7 17 46 53 126	2 1 7 3 0 0 0 0	170 108 0 122 0
6 25 31 50 73 74	8 18 47 54 127	4 7 10 2 12 5 0 0 0	144 67 30 55 0
3 5 27 53 74 75	0 9 19 48 128	1 1 1 2 4 2 7 0 0 0	130 55 0 11 0
2 4 28 30 52 75 76	1 10 20 49 129	3 3 1 5 9 6 1 7 8 0 0 0	20 83 138 53 0
1 8 32 54 76 77	0 11 21 50 130	1 5 1 2 6 1 8 0 0 0	8 3 3 9 9 9 1 1 5 0
12 35 40 49 77 78	1 1 2 2 2 5 1 1 3 1	1 7 4 1 1 0 2 4 1 0 7 0 0	1 4 1 7 1 1 1 8 1 2 6 0
11 13 14 41 78 79	0 1 3 2 3 5 2 1 3 2	1 0 5 3 1 8 9 8 4 0 0	7 2 1 3 6 1 0 0 0 0
0 3 3 3 7 5 0 5 5 7 9 8 0	1 1 4 2 4 5 3 1 3 3	3 5 1 4 7 7 0 1 7 0 0 0	1 3 3 1 2 8 2 3 0 0
10 3 4 3 9 4 8 8 0 8 1	0 1 5 2 5 5 4 1 3 4	4 1 2 1 9 2 1 2 3 0 0	1 4 3 1 4 6 4 6 0 0
1 1 6 1 8 5 1 8 1 8 2	1 1 6 2 6 4 2 1 3 5	1 2 7 1 0 1 2 1 6 5 0 0	1 5 3 9 1 1 0 0 0 0
1 5 1 7 3 6 3 8 8 2 8 3	0 1 7 2 7 4 3 1 3 6	7 4 1 5 1 9 4 1 1 0 0	1 3 1 4 6 1 8 1 7 0
1 0 2 3 2 5 4 5 8 3 8 4	1 1 8 2 8 4 4 1 3 7	1 4 1 1 2 6 1 6 7 1 7 1 0 0	1 7 5 1 5 2 1 0 9 1 2 0 0
2 2 4 4 2 5 2 8 4 8 5	0 1 9 2 9 4 5 1 3 8	3 7 6 2 3 1 7 3 0 0	1 6 9 0 6 3 1 7 5 0
1 1 9 2 2 4 7 8 5 8 6	1 2 0 3 0 4 6 1 3 9	1 4 6 1 6 6 9 6 5 4 0 0	6 4 9 2 5 8 0 0
2 1 3 4 4 4 5 3 8 6 8 7	0 2 1 3 1 4 7 1 4 0	1 4 3 1 5 1 8 6 1 7 0 0 0	1 7 0 0 1 2 7 0 0
2 0 2 6 3 3 4 6 8 7 8 8	1 2 3 2 4 8 1 4 1	9 1 8 7 0 8 1 0 0	1 6 4 8 4 8 0 0
3 1 1 4 3 5 4 8 8 8 9	0 2 2 2 1 4 2	6 7 1 3 9 6 1 5 7 0 0	1 6 6 1 4 7 0 0
5 1 3 1 5 4 2 8 9 9 0	1 3 2 3 1 4 3	7 6 1 3 3 1 1 0 1 5 6 0 0	2 6 3 3 0 0
1 2 2 9 3 6 4 8 9 0 9 1	0 4 2 4 1 4 4	8 5 4 1 4 5 1 5 5 0 0	2 2 2 5 0 0
4 2 8 4 9 5 2 9 1 9 2	1 5 2 5 1 4 5	6 5 1 2 2 3 5 8 1 0 0	1 6 9 1 0 7 0 0
6 1 9 2 1 4 3 9 2 9 3	0 6 2 6 1 4 6	6 0 1 1 3 1 1 9 1 6 4 0 0	1 6 9 0 6 5 0
2 7 3 7 4 4 5 3 9 3 9 4	1 7 2 7 1 4 7	9 2 4 6 0 0	1 5 6 7 6 0 0
0 1 4 2 0 3 5 9 4 9 5	0 8 2 8 1 4 8	1 5 2 1 0 3 6 1 1 7 0 0	1 3 2 4 6 0 0
7 2 2 2 3 4 6 9 5 9 6	1 9 2 9 1 4 9	1 5 6 6 1 7 2 1 1 0 0	2 8 9 3 0 0
1 7 3 0 3 9 5 1 9 6 9 7	0 1 0 3 0 1 5 0	1 7 4 3 9 1 3 1 1 5 6 0 0	1 0 8 0 0 0
9 1 6 2 4 3 8 9 7 9 8	1 1 1 3 1 1 5 1	1 4 6 1 6 2 9 4 3 0 0	1 3 4 0 6 1 0
0 8 4 0 5 4 9 8 9 9	0 1 2 3 2 1 5 2	5 9 5 7 1 6 2 8 6 0 0	2 3 0 0 0
1 3 1 3 2 4 7 9 9 1 0 0	1 1 3 3 3 1 5 3	1 1 7 1 6 9 7 7 1 0 3 0 0	1 1 5 4 0 0 0
1 8 4 1 4 5 5 0 5 5 1 0 0	0 1 4 3 4 1 5 4	9 3 8 0 1 5 8 7 1 1 0	1 3 9 0 0 0
0 2 1 6 1 0 1	1 1 5 3 5 1 5 5	4 8 1 5 6 0 0	8 4 6 0 0 0
1 3 1 7 1 0 2	0 1 6 3 6 1 5 6	1 2 6 1 5 0 0	1 6 8 0 0 0
0 4 1 2 1 8 1 0 3	1 1 7 3 7 1 5 7	2 6 6 8 1 0 0 0	1 0 3 4 0 0 0
1 5 1 3 1 9 1 0 4	0 1 8 3 8 1 5 8	7 1 2 1 6 8 3 3 0	1 0 0 9 2 4 2 0
0 6 1 4 2 0 1 0 5	1 1 9 3 9 1 5 9	9 4 3 4 0 3 7 0	1 0 2 3 0
1 7 1 5 2 1 1 0 6	0 2 0 4 0 1 6 0	1 4 2 1 2 1 7 0 0	4 6 0 0 0
0 2 8 1 6 1 0 7	1 2 1 4 1 1 6 1	1 3 7 9 7 1 0 6 0 0	1 3 1 1 5 4 0 0
1 3 9 1 7 1 0 8		9 8 7 6 3 5 0 0	











